REMARKS

Claims 1, 3, 5-9, 11-16, 18-21, and 23-31 are pending. Claims 9 and 21 have been amended.

Upon review, Applicant has noticed that initialed copies of the PTO Form 1449's filed on May 22, 2003 and September 16, 2004 are not present in his records. Applicant respectfully requests that the Examiner consider the references cited therein and return the initialed PTO Form 1449's with the next Office action.

If the PTO Form 1449's and/or the references have been misplaced, the Examiner is invited to telephone the undersigned and copies will be forwarded.

Rejections under 35 U.S.C. § 102(b)

Claim 9

In the action mailed December 21, 2004, claim 9 was rejected under 35 U.S.C. § 102(b) as anticipated by U.S. Patent No. 5,970,241 to Deao et al. (hereinafter "Deao").

As amended, claim 9 relates to a method of providing instructions to a processor. The method includes loading a plurality of instructions into an emulation instruction register, receiving a run-test idle state signal, providing the plurality of instructions to the processor in response to the receipt of the run-test idle state signal, and processing the

plurality of instructions without receiving another run-test idle state signal.

The rejection of claim 9 contends that Deao describes the provision of a plurality of instructions to a processor in response to the receipt of the run-test idle state signal. Applicant respectfully disagrees.

Deao describes a microprocessor that includes an emulation unit for debugging the operation of the microprocessor's instruction execution pipeline. In Deao, a "pipe down" procedure is used when the instruction execution pipeline is stopped. The pipe down procedure saves the contents of various registers within the pipeline so that debugging can occur. *See* col. 5, line 25-27 and col. 45, line 18-21. As part of the pipe down procedure, the fetching of instructions from memory is stopped. *See* col. 5, line 30-34 (describing that the subsystem data pipeline is halted in response to receipt of a signal indicating that the processor pipeline has halted) and col. 44, line 37-52 and col. 45, line 30-34 (describing that after a halt is initiated, program fetching is halted with EPDN set to 1).

Since program fetching is halted in Deao's "pipe down" procedure, Deao neither describes nor suggests the provision of a plurality of instructions to a processor in response to the receipt of the run-test idle state signal. Instead, Deao stops the flow of instructions to Deao's processor when the pipe down

procedure is initiated. Such a stoppage of the flow of instructions has nothing to do with the provision of instructions in response to receipt of a run-test idle state signal.

Accordingly, Applicant submits that claim 9, and the claims dependent therefrom, are patentable over Deao.

Claim 16

Claim 16 was rejected under 35 U.S.C. § 102(b) as anticipated by Deao.

Claim 16 relates to a processor that includes a test interface, an emulation instruction register adapted to store a plurality of emulation instructions received from the test interface, emulation control logic adapted to control a flow of the plurality of emulation instructions to a processor pipeline following detection of a single run-test idle state, and a decoder to receive the plurality of instructions for processing.

The rejection of claim 16 points to col. 5, line 12-38 of Deao as allegedly describing emulation control logic adapted to control a flow of the plurality of emulation instructions to a processor pipeline following detection of a single run-test idle state. Applicant respectfully disagrees.

Col. 5, line 30-34 of Deao describes that a subsystem data pipeline is halted in response to receipt of a signal indicating that the processor pipeline has halted. The halting of the data

pipeline and the processor pipeline stop program fetching. *See* col. 44, line 37-52 and col. 45, line 30-34 (describing that after a halt is initiated, program fetching is halted).

Since program fetching is halted following detection of a pipeline halt state, Deao neither describes nor suggests emulation control logic adapted to control a flow of the plurality of emulation instructions to a processor pipeline following detection of a single run-test idle state. Indeed, Deao *stops* the flow of instructions to Deao's processor when a pipeline halt is initiated. With the flow of instructions stopped, there is no flow of emulation instructions to a processor pipeline to be controlled.

Accordingly, Applicant submits that claim 16, and the claims dependent therefrom, are patentable over Deao.

Claim 21

Claim 21 was rejected under 35 U.S.C. § 102(b) as anticipated by Deao.

As amended, claim 21 relates to an apparatus that includes operating instructions residing on a machine-readable storage medium. The operating instructions are for use in a device to handle a plurality of emulation instructions. The operating instructions cause the device to load the plurality of emulation instructions into a single emulation instruction register, enter a run-test idle state, provide the plurality of emulation

instructions to a processor in response to entry into the run-test idle state, and process the plurality of emulation instructions.

The rejection of claim 21 points to numerous portions of Deao as allegedly describing the provision of a plurality of emulation instructions to a processor in response to entry into the run-test idle state. However, each of these portions relates to the normal operation of Deao's system and the emulation instructions are not provided in response to entry into a run-test idle state.

As discussed above, when Deao's instruction execution pipeline is stopped, Deao's system executes a pipe down procedure in which the fetching of instructions from memory is *stopped*. Such a stoppage of the flow of instructions has nothing to do with the provision of emulation instructions in response to entry into a run-test idle state.

Accordingly, Applicant submits that claim 21, and the claims dependent therefrom, are patentable over Deao.

Rejections under 35 U.S.C. § 103(a)

Claim 1

Claim 1 was rejected under 35 U.S.C. § 103(a) as obvious over Deao and U.S. Patent No. 5,848,288 to O'Connor (hereinafter "O'Connor").

Claim 1 relates to a method that includes receiving a plurality of instructions from a test interface, loading the plurality of instructions into an emulation instruction register; receiving a plurality of instructions from the emulation instruction register, determining a validity of a first instruction of the plurality of instructions by reading width bits in the first emulation instruction, providing the first instruction to a decoder of the processor if the first instruction is valid; without receiving a run-test idle state signal, determining a validity of a second instruction of the plurality of instructions by reading width bits in the second instruction, and providing the second instruction to the decoder if the second instruction is valid.

The rejection of claim 1 contends that O'Connor describes determining the validity of an instruction by reading width bits in the instruction.

As support for this contention, the rejection contends that O'Connor's "predetermined bit" that is set to indicate an end of an instruction bundle constitutes "width bits" in an

instruction. The rejection also contends that the validity of an instruction bundle is somehow determined by O'Connor's identifying this bit.

Applicant disagrees with both contentions. O'Connor describes a computer system in which variable size instruction bundles exist. *See* col. 2, line 5-6. The instruction bundles are not necessarily the same size as a single fetch. Rather, fetched instructions are partially decoded to determine if an end of bundle exists in a particular set of fetched instructions. *See* col. 2, line 8-10. An end of *a bundle of instructions* is denoted by a predetermined bit, or implicitly by the nature of the instructions themselves. *See* col. 2, line 65 - col. 3, line 15.

If an end of a bundle is found in the partially decoded set of instructions, then instructions after the end of the bundle are prevented from issuing. *See* col. 2, line 10-12. This is done to allow software developed for machines that fetch relatively high numbers of instructions per cycle to be performed on machines that fetch lower numbers of instructions per cycle. *See* col. 1, line 60 - col. 2, line 2.

Applicant submits that O'Connor's "predetermined bit" is not width bits in the instruction. Width bits define the validity and size of an instruction. *See* page 13, line 21-23 of the specification. Since O'Connor's bit only demarks *the end of*

a variable size instruction bundle, O'Connor's bit is not found *in* the instruction, nor has it anything to do with either the validity or the size of an instruction. In particular, an identification of the end of a bundle does not define the size of any of the bundle's constituent instructions nor does it provide any indication of whether the constituent instructions are in fact valid.

The fact that O'Connor's bit does not define whether or not a constituent instruction is valid is reflected in the use that O'Connor makes of that bit. In particular, O'Connor neither describes nor suggests using the bit to determine the validity of an instruction. Rather, O'Connor simply uses the bit to prevent the issuance of instructions after the end of a bundle. The validity of the instructions- either before or after the bit- is never checked.

As discussed in the rejection, Deao neither describes nor suggests determining the validity of an instruction by reading width bits in the instruction. Thus, Deao does nothing to remedy this deficiency in O'Connor.

Accordingly, Applicant submits that claim 1, and the claims dependent therefrom, are patentable over Deao and O'Connor, alone or in combination.

Applicant asks that all claims be allowed. No fees are believed due at this time. Please apply any charges or credits to Deposit Account No. 06-1050.

Respectfully submitted,

Date: 2/10/05

Scott C. Harris
Reg. No. 32,030

BY
JOHN F. CONROY
REG # 45,485

Fish & Richardson P.C.
PTO Customer Number: 20985
12390 El Camino Real
San Diego, CA 92130
Telephone: (858) 678-5070
Facsimile: (858) 678-5099
10477043.doc